

# Penerapan *Intelligence System* berbasis *Case Base Reasoning* dan Metode *K-Nearest Neighbor* Untuk Identifikasi Gangguan IT Support

Muhamad Dedi Suryadi<sup>1\*)</sup>, Sahlan<sup>2</sup>, Ndaru Ruseno<sup>3</sup>

<sup>1,2,3</sup> Jurusan Teknik Informatika, STMIK Bani Saleh, Bekasi

<sup>1,2,3</sup>Jl. M. Hasibuan No. 68, Kota Bekasi, 17113, Indonesia

email: <sup>1</sup>kangededi@gmail.com, <sup>2</sup>sahlan.c19@gmail.com, <sup>3</sup>ndarumantap@gmail.com

*Special Issue on Seminar Nasional - Inovasi Dalam Teknologi Informasi & Teknologi Pembelajaran 2019*

**Abstract** - Delay in handling and solving IT problems both hardware / software greatly affects the company's business processes. Sometimes, IT helpdesk personnel still difficult to find solutions and make decisions to solve IT problems. This is because there is currently no system that can help IT Helpdesk personnel in finding solutions to be able to solve the problems being faced quickly and accurately. When going to look for solutions to the problems being faced can not display solutions that are appropriate or close to the existing historical data. In identifying IT problems, the authors use case-based reasoning or Case-Based Reasoning (CBR) by carrying out the process of finding the case with the highest proximity and proximity measurement using the K-Nearest Neighbor (k-NN) algorithm. This study aims to explain the application of Case-Based Reasoning (CBR) and K-Nearest Neighbor (k-NN) Algorithm to identify IT problem disturbances. The results showed that the application of Case-Based Reasoning and K-Nearest Neighbor (k-NN) Algorithm after being tested by 5 Experts got accurate results.

**Keywords** - case based reasoning, k-nearest neighbor algorithm, help desk

**Abstrak** – Keterlambatan dalam menangani dan menyelesaikan permasalahan IT baik hardware/software sangat berdampak terhadap proses bisnis perusahaan. Terkadang, personel IT helpdesk masih kesulitan dalam mencari solusi serta mengambil keputusan untuk dapat menyelesaikan permasalahan IT. Hal ini dikarenakan saat ini belum terdapat sebuah sistem yang dapat membantu personel IT Helpdesk dalam mencari solusi untuk dapat menyelesaikan permasalahan yang sedang dihadapi secara cepat dan akurat. Ketika akan mencari solusi dari permasalahan yang sedang dihadapi belum bisa menampilkan solusi yang sesuai ataupun yang mendekati dari historical data yang sudah ada. Didalam mengidentifikasi permasalahan IT, penulis menggunakan penalaran berbasis kasus atau Case-Based Reasoning (CBR) dengan melakukan proses pencarian kasus yang memiliki kedekatan paling tinggi dan pengukuran kedekatan dengan menggunakan algoritma K-Nearest Neighbor (k-NN). Penelitian ini bertujuan memaparkan penerapan Case-Based Reasoning (CBR) dan Algoritma K-Nearest Neighbor (k-NN) untuk identifikasi gangguan permasalahan IT. Hasil penelitian menunjukkan bahwa penerapan Case-Based Reasoning dan Algoritma K-Nearest Neighbor (k-NN) setelah diujikan oleh 5 orang Pakar mendapatkan hasil yang akurat.

**Kata Kunci** – case base reasoning, algoritma k-nearest neighbor, helpdesk

## I. PENDAHULUAN

Keterlambatan dalam menangani dan menyelesaikan permasalahan IT baik *hardware/software* yang terjadi di PT. Pasifik Satelit Nusantara, sangat berdampak terhadap proses bisnis perusahaan. Terkadang, personel IT *helpdesk* masih kesulitan dalam mencari solusi serta mengambil keputusan untuk dapat menyelesaikan permasalahan IT. Hal ini dikarenakan saat ini belum terdapat sebuah sistem yang dapat membantu personel IT Helpdesk dalam mencari solusi untuk dapat menyelesaikan permasalahan yang sedang dihadapi secara cepat dan akurat. solusi dari setiap permasalahan IT saat ini masih berupa *record-record* dalam *database*. Ketika akan mencari solusi dari permasalahan yang sedang dihadapi belum bisa menampilkan solusi yang sesuai ataupun yang mendekati dari *historical* data yang sudah ada.

*Case-Based Reasoning* (CBR) merupakan metode yang dapat digunakan untuk memecahkan masalah dengan menghitung kedekatan/kemiripan antara kasus baru dengan kasus lama. Kasus yang memiliki kemiripan tertinggi atau yang paling mirip akan digunakan sebagai solusi untuk memecahkan masalah yang baru [1] [2]. Metode CBR ini cocok diterapkan untuk menangani *complain* karena metode ini dapat digunakan untuk mencari solusi dari kasus yang pernah terjadi dengan menghitung nilai *similarity* tertinggi atau yang paling mirip akan digunakan sebagai solusi untuk memecahkan masalah yang baru.

Dengan diterapkannya metode *case based reasoning* dan algoritma *K-Nearest Neighbor* untuk sistem penanganan gangguan dan *maintenance* IT pada PT. Pasifik Satelit Nusantara maka diharapkan dapat meningkatkan efektivitas pelayanan terhadap keluhan yang dilaporkan karyawan agar dapat diselesaikan dengan cepat dan akurat.

## II. PENELITIAN YANG TERKAIT

Penelitian yang dilakukan Sandy Kosasi [3] dengan judul pembuatan aplikasi diagnosa kerusakan mesin sepeda motor matic dengan *case-based reasoning*. Kesimpulan dari

\*) penulis korespondensi: Muhamad Dedi Suryadi  
Email: kangededi@gmail.com

penelitian ini membangun sistem pakar menggunakan *Case Base Reasoning* untuk melakukan identifikasi kerusakan mesin sepeda motor matic dengan memberikan solusi penyelesaian masalah dengan rata-rata nilai similaritas 0,62 dan 0,7 dengan nilai keakuratan solusi dari pakar sebesar 80% sampai 90%.

Penelitian yang dilakukan Moh Zainudin, Khasnur Hidjah dan I Wayan Tunjung [4] terkait dengan penerapan *case base reasoning* (cbr) untuk mendiagnosis penyakit stroke menggunakan algoritma *k-nearest neighbor*. Kesimpulan dari penelitian tersebut menyimpulkan bagaimana menerapkan sistem pakar menggunakan *Case Base Reasoning* untuk mendiagnosis penyakit stroke dengan menggunakan algoritma *K-Nearest Neighbor* dengan menghitung nilai similarity Antara kasus baru dengan kasus lama sehingga kasus baru didapat nilai kemiripan diagnosis sebesar 93.3%.

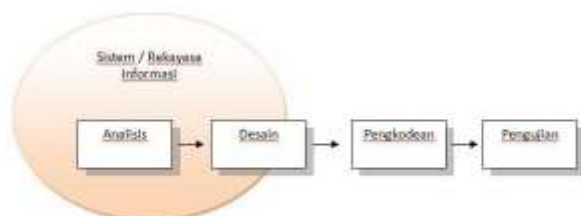
Penelitian yang dilakukan oleh Wahyuni Rodiyah Risfianti, Tacbir Hendro Pudjiantoro dan Asep Id Hadiana [5] mengenai penentuan penanganan kasus terhadap penyakit berdasarkan gejala menggunakan *Case Base Reasoning* dan algoritma *Nearest Neighbor* studi kasus: Klinik Citra Medika Cianjur. Kesimpulan dari penelitian ini membangun sistem pakar yang dapat membantu tenaga medis dalam menentukan penanganan kasus bagi pasien berdasarkan rekam medis dan gejala penyakit menggunakan *Case Base Reasoning* dan algoritma *Nearest Neighbor*, dimana sistem mampu mencari kemiripan antara kasus lama yang ada pada basis kasus dengan kasus baru yang dicari berdasarkan perhitungan dari data yang telah diuji.

Penelitian yang dilakukan oleh Minarni, Indra Warman, Wenda Handayani [6] mengenai *case-based reasoning* (CBR) pada sistem pakar identifikasi hama dan penyakit tanaman singkong dalam usaha meningkatkan produktivitas tanaman pangan. Hasil dari penelitian ini adalah membangun sistem pakar untuk melakukan identifikasi hama dan penyakit pada tanaman singkong dengan menggunakan algoritma *Nearest Neighbor* dengan similaritas Antara kasus baru dengan kasus lama sehingga hasil uji sistem mampu mengidentifikasi hama dan penyakit tanaman dengan gejala sesuai rule dengan tingkat akurasi sebesar 67,65%.

### III. METODE PENELITIAN

#### A. Tahapan Penelitian

Penelitian ini menggunakan metode *waterfall* [7][8]. *Waterfall* adalah pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian.



Gbr. 1 Model Waterfall

Fase Analisis, dalam fase ini penulis mengumpulkan informasi mengenai kebutuhan atribut dan parameter yang

akan digunakan. Dalam penelitian ini penulis mengambil empat parameter sebagai data latih untuk perhitungan algoritma *K-Nearest Neighbor*, diantaranya yang pertama adalah parameter kategori, yang kedua adalah parameter subkategori, yang ketiga adalah parameter indikasi primer (*primary*) dan yang keempat adalah parameter indikasi sekeunder (*secondary*) dengan masing-masing bobot atribut seperti tabel I, Tabel II, Tabel III, Tabel IV, dan Tabel V.

TABEL I  
ATRIBUT DAN BOBOT PELATIHAN ALGORITMA K-NN

Atribut	Nilai
Kategori Gangguan	0.8
Subkategori Gangguan	0.7
Indikasi Primer	0.6
Indikasi Sekunder	0.5

TABEL II  
ATRIBUT KATEGORI GANGGUAN

NO	KATEGORI	KODE
1	Software	CAT01
2	Hardware	CAT02
3	Network	CAT03

TABEL III  
ATRIBUT SUBKATEGORI GANGGUAN

NO	SUBKATEGORI	KODE
1	Set PC bermasalah	SK01
2	Masalah pada aplikasi pendukung kerja	SK02
3	Masalah pada aplikasi internal	SK03
4	Set printerproblem	SK04
5	Masalah konfigurasi PC	SK05

TABEL IV  
ATRIBUT INDIKASI PRIMER

NO	INDIKASI PRIMER	KODE
1	Aplikasi <i>cbossnot responding</i>	IP01
2	Aplikasi <i>email not responding</i>	IP02
3	Aplikasi <i>email</i> tidak bisa menerima pesan	IP03
4	Aplikasi <i>email</i> tidak bisa mengirim pesan	IP04
5	Aplikasi <i>MS. Office not responding</i>	IP05

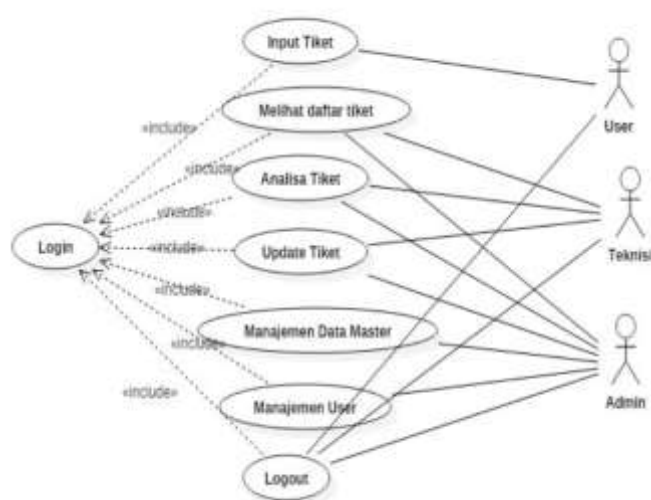
TABEL V  
ATRIBUT INDIKASI SEKUNDER

NO	INDIKASI SEKUNDER	KODE
1	Aplikasi <i>email</i> muncul notifikasi <i>connection to server time out</i>	IS01
2	Aplikasi <i>email</i> muncul notifikasi <i>failed connect to server</i>	IS02
3	Aplikasi <i>email thunderbird full inbox</i>	IS03
4	Aplikasi <i>mail client</i> tidak bisa dijalankan	IS04
5	PC Bootlooping	IS05

Bobot nilai *similarity* merupakan bobot yang didapat dengan menghitung kedekatan atau kemiripan nilai-nilai didalam atribut.

TABEL VI  
BOBOT NILAI SIMILARITY

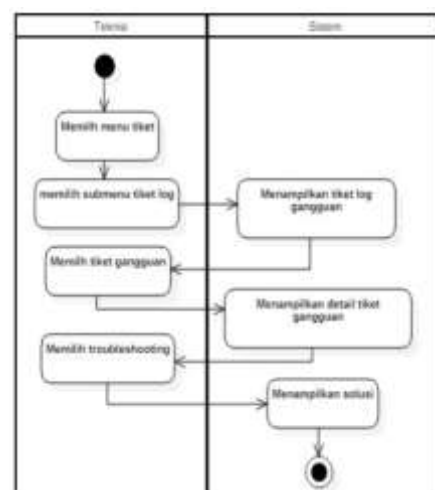
NO	Nilai Atribut 1	Nilai Atribut 2	Bobot
1	Software	Software	1
2	Software	Hardware	0
3	Software	Network	0.5
4	Hardware	Software	0
5	Hardware	Hardware	1
6	Hardware	Network	0.5
7	Network	Software	0.5
8	Network	Hardware	0.5
9	Network	Network	1



Gbr. 2 Use Case Diagram

Fase *Design*, dalam fase design penulis menggunakan *Use Case Diagram* dan *Activity Diagram*. *Use Case* digunakan untuk menggambarkan interaksi antara pengguna program (aktor) dengan kasus (*use case*) yang disesuaikan dengan langkah-langkah yang telah ditentukan. Pada Gbr.2 adalah use case diagram identifikasi gangguan IT

Pada activity diagram seperti pada Gbr.3 dijelaskan proses dimana teknisi melakukan analisa permasalahan untuk mengetahui solusi penyelesaian berdasarkan data tiket yang telah di input oleh *user*. Fase pengkodean, pada fase ini penulis menggunakan bahasa pemrograman php, sublime, xampp. Fase pengujian, pada fase ini menggunakan metode *blackbox*. Menurut [9] *Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak

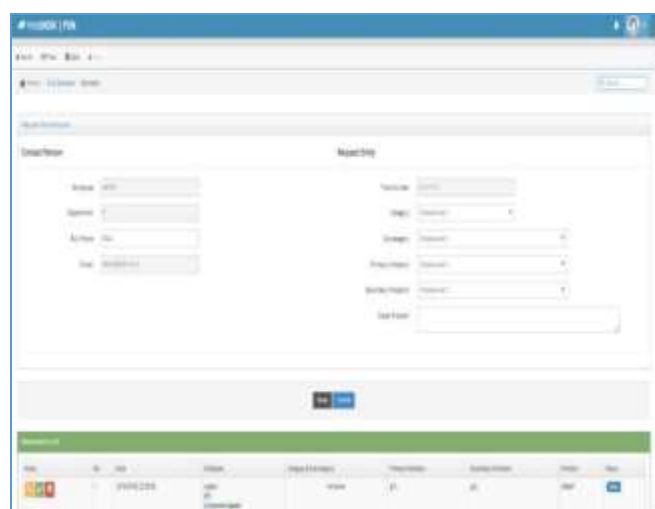


Gbr. 3 Activity Diagram

## IV. HASIL DAN PEMBAHASAN

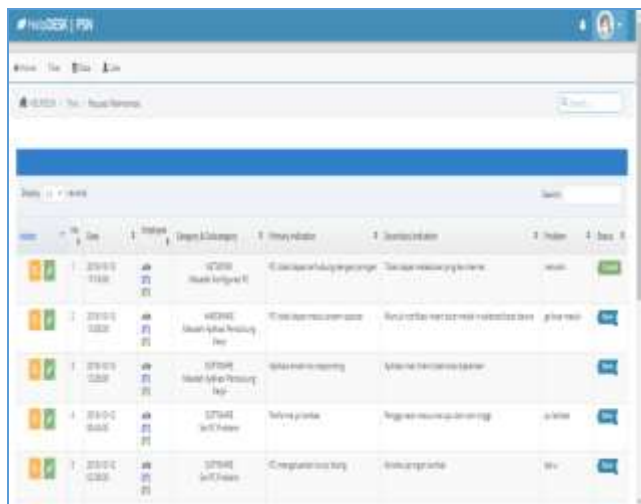
### A. Implementasi Sistem

Ketika terjadi gangguan IT, *user* dapat menyampaikan gangguan pada tim IT-support dengan membuat tiket pengaduan pada sistem dengan mengisi data-data indikasi gangguan. Setelah data tiket diinput oleh user, maka pada menu daftar tiket akan terdapat tiket baru, yang berisi data-data indikasi gangguan. Untuk dapat melakukan identifikasi teknisi harus mengklik tombol analisa pada salah satu tiket, seperti pada Gbr.5.



Gbr. 4 Tampilan Menu *Create Tiket*

Form analisa tiket berfungsi untuk menampilkan hasil solusi dari permasalahan yang dilaporkan dengan mencari solusi berdasarkan kasus-kasus yang pernah terjadi, atau dengan menghitung nilai kemiripan tertinggi antara kasus lama dengan kasus baru dengan menggunakan algoritma *K-Nearest Neighbor* (k-NN) yang kemudian akan ditindaklanjuti untuk proses penyelesaian gangguan IT-support. Setelah tiket dianalisa maka akan muncul form hasil dari analisa tiket gangguan, seperti diperlihatkan pada Gbr.6.



Gbr. 5 Tampilan Menu Daftar Tiket



Gbr. 6 Tampilan Hasil Analisa Tiket



Gbr. 7 Algoritma identifikasi gangguan IT-Support

Pada Gambar 7 adalah alur dari identifikasi gangguan IT-Support. Dimulai dari proses pengambilan data tiket, selanjutnya system akan mencari kasus yang mirip dari repository, jika kasus yang serupa tidak ditemukan, maka akan menghitung nilai kemiripan dengan mencari nilai kemiripan yang tertinggi. Jika nilai kemiripan yang tertinggi ditemukan, maka nilai kemiripan yang tertinggi akan ditampilkan sebagai solusi dari penanganan.

### B. Akurasi

Pengujian akurasi dilakukan untuk mengetahui performa dari sistem pakar untuk memberikan hasil identifikasi permasalahan yang dilaporkan oleh karyawan. Data yang diuji berjumlah 5 sampel data analisa pakar. Hasil pengujian diperlihatkan pada Tabel VII.

TABEL VII  
PENGUJIAN AKURASI

NO	Data Sample	Akurasi
1	Category : hardware Subcategory : masalah aplikasi pendukung kerja Indikasi primer : pc tidak dapat masuk system operasi Indikasi sekunder : muncul notifikasi insert boot media in selected boot device Solusi : periksa pemasangan kabel power dan data pada harddisk	[Ya]
2	Category : software Subcategory : set pc problem Indikasi primer : performa pc lambat Indikasi sekunder : penggunaan resource cpu dan ram tinggi Solusi : Periksa service aplikasi yang running, matikan service aplikasi yang tidak diperlukan.	[Ya]
3	Category : hardware Subcategory : set printer problem Indikasi primer : hasil print tidak keluar Indikasi sekunder : paper jam Solusi : pada printer, buka bagian penutup toner, keluarkan toner terlebih dahulu, keluarkan kertas yang tersangkut didalam printer, masukan kembali toner dan tutup kembali cover penutup	[Ya]
4	category : software subcategory : masalah aplikasi pendukung kerja indikasi primer : aplikasi email tidak dapat menerima pesan indikasi sekunder : tidak dapat melakukan ping ke intranet Solusi : Periksa koneksi jaringan dari PC atau Laptop ke jaringan internetdengan	[Ya]
5	Category : software subcategory : masalah aplikasi pendukung kerja indikasi primer : aplikasi email tidak dapat menerima email indikasi sekunder : aplikasi email thunderbird full inbox Solusi : masuk pada menu account setting, backup terlebih dahulu email inbox yang lama dan kemudian create ulang mail inbox yang baru	[Ya]

## V. KESIMPULAN

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa: Proses penanganan gangguan yang dilakukan oleh bagian IT-support menjadi lebih efektif dan efisien sehingga tidak terlalu banyak waktu yang terbuang. Proses identifikasi gangguan menjadi lebih cepat dan akurat dengan menggunakan *Intelligence System* Berbasis *Case-Based Reasoning* dengan Algoritma *K-Nearest Neighbor*. Sistem dapat memberikan solusi penyelesaian secara cepat dan akurat sehingga membantu teknisi dalam proses identifikasi gangguan IT.

## UCAPAN TERIMA KASIH

Penulis mengucapkan Terima kasih kepada semua pihak baik pihak STMIK Bani Saleh, PT. PSN ataupun pihak-pihak lain yang sudah membantu penelitian ini.

## DAFTAR PUSTAKA

- [1] I. Watson and F. Marir, "Case-based reasoning: A review," *Knowl. Eng. Rev.*, vol. 9, no. 4, pp. 327–354, Dec. 1994, doi: 10.1017/S0269888900007098.
- [2] Kusrini and E. T. Luthfi, *Algoritma data mining*. Yogyakarta: Andi Offset, 2009.
- [3] S. Kosasi, "Pembuatan Aplikasi Diagnosa Kerusakan Mesin Sepeda Motor Matic dengan Case-Based Reasoning," *Creat. Inf. Technol. J.*, vol. 2, no. 3, p. 192, Jun. 2015, doi: 10.24076/citec.2015v2i3.48.
- [4] M. Zainuddin, K. Hidjah, and I. W. Tunjung, "PENERAPAN CASE BASED REASONING(CBR) UNTUK MENDIAGNOSIS PENYAKIT STROKE MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR," in *CITISEE 2016*, 2016, pp. 21–26.
- [5] W. R. Risfianti, T. H. Pudjiantoro, and A. I. Hadiana, "Penentuan Penanganan Kasus Terhadap Penyakit Berdasarkan Gejala Menggunakan Case Base Reasoning Dan Algoritma Nearest Neighbor (Studi Kasus: Klinik Citra Medika Cianjur)," in *Prosiding SNST Fakultas Teknik*, 2016.
- [6] M. Minami, I. Warman, and W. Handayani, "Case-Based Reasoning (Cbr) Pada Sistem Pakar Identifikasi Hama Dan Penyakit Tanaman Singkong Dalam Usaha Meningkatkan Produktivitas Tanaman Pangan," *J. Teknoif*, vol. 5, no. 1, 2017.
- [7] R. AS and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berbasis Objek*. Bandung: Informatika Bandung, 2015.
- [8] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th*. McGraw-Hill Education, 2010.
- [9] T. S. Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 45–48, 2018, doi: 10.30591/jpit.v3i1.647.